

A sparse recovery method for initial ranging in IEEE 802.16 OFDMA systems

Md Mashud Hyder and Kaushik Mahata

Abstract—Initial ranging constitutes a part of the synchronization procedure employed by the wireless communication standards. This allows the base station (BS) to detect the subscriber stations (SS) that are willing to commence communication. In addition, the ranging process allows the BS to estimate the uplink channel parameters of these SSs. Accurate estimation of these parameters are crucial as they ensure that the uplink signals from all the SSs arrive at the BS synchronously and approximately at the same power level. However, this detection and estimation problem turns out to be very challenging when multiple users initiate the ranging procedure at the same time. We address this issue by exploiting the underlying sparsity of the estimation problem. We propose a fast sparse signal recovery approach to improve the ranging performance in multi-user environment. Compared to the standard correlation based techniques, our method shows a clear improvement in ranging code detection, timing offset and channel power estimation. Although this method has been developed around the WiMAX standard, the underlying principles apply to other OFDM based standards as well.

Index Terms—Initial ranging, code detection, sparse representation, OFDMA.

I. INTRODUCTION

The orthogonal frequency-division multiple access (OFDMA) scheme has been adopted by the IEEE WiMAX standard [1]. To maintain orthogonality among the subcarriers of different users, and to avoid the occurrence of multiple-access interference (MAI), the uplink signals arriving at the BS must be aligned to the local time and frequency references. For this purpose, the IEEE WiMAX standard enforces a network entry procedure called initial ranging (IR). The ranging process starts with the allocation of a pre-defined set of subcarriers by the BS in specific time slots, which is known as a ranging opportunity. The SSs which wish to commence communication with the BS, referred to as the ranging terminals (RTs), can take this opportunity by modulating a randomly selected ranging code onto the allocated subcarriers. Due to different position of RTs within the radio coverage area, ranging signals transmitted by different RTs arrive at the BS with their specific transmission time delay. At the receiving side, the BS is required to extract timing and power information for each detected code and inform the RTs about the extracted information.

Multiuser code separation as well as their timing and power estimation are the main tasks of the IR process [2]. The correlation based approach proposed in [3], is based on the principle

that a time delay can be represented by a phase shift in the frequency domain. Lee [4] replaces the WiMAX ranging codes by a set of generalized chirp-like polyphase sequences to get a more accurate timing estimate. The work in [5] demonstrates that the frequency-domain correlation approach outperforms its time domain counterpart. However, the methods [3]–[5] simply treat the MAI as a noise, which results in performance degradation in multiuser environment. An approach different from the IEEE 802.16 standards has been proposed in [6] to overcome the MAI problem in code detection. The idea is to allocate a small number of subcarriers to each ranging opportunity so that most of the RTs are expected to transmit on disjoint sets of subcarriers resulting a minimum level of MAI. However, the reduction of the number of effective subcarrier for each user results in the degradation of timing estimation performance [2]. A similar approach has been proposed in [7] for channel synchronization. This method assumes that the uplink signals are transmitted over disjoint subcarriers, and the receivers use filter banks to separate multiuser codes. The work in [8] improves the ranging performance by dividing the ranging signals into several groups with each group being transmitted over exclusively assigned subcarriers. The concept of successive interference cancellation (SIC) has been employed in [2], [9]. The method proposed in [2] works in an iterative fashion where the strongest path of each active RT is detected and is removed from the received signal and the resulting signal is used in succeeding iterations. The work [9] differs from [2] in the aspect is that in [9] one additional ranging signal is detected at each iteration instead of a single multipath component.

Sparse signal representation [10], [11] has many potential applications, including spectral analysis [12], channel estimation [13] etc. In this paper, we pose the problem of user detection and channel estimation in ranging process as a sparse recovery problem. This approach is founded on the two observations:

- The number of ranging subscriber stations is much smaller than the number of ranging codes.
- Only a few channel taps are of some noticeable magnitude, and the remaining vast majority of channel taps are of negligible magnitude.

We first develop a signal model that allows us to exploit the above facts and pose the ranging problem as a sparse recovery problem. By applying the standard sparse recovery methods to solve the sparse recovery problem arising in ranging problem, we found that associated computation time is too high compared to what is typically allowed by the WiMAX

Department of Electrical Engineering, The University of Newcastle, Australia.

Research is supported by the Australian Research Council.

standard. For this reason, we propose a new algorithm to solve the sparse-ranging problem. The proposed algorithm combines two different types of sparse recovery algorithms to provide a time efficient solution. In particular, we apply a non-convex sparse recovery algorithm which has fast convergence property if it initialized sufficiently close to the final solution. To obtain a good initial input, we consider another convex optimization algorithm that can provide a good estimate of the final solution after a few iterations. The proposed handover algorithm computes most of its multiplications by using Fast Fourier Transform (FFT). Due to noise contribution in the ranging signal, the recovered signal from the algorithm may not be truly sparse. For this reason, we analyse the statistical property of the recovery error. This analysis is used to formulate a systematic hypothesis test to detect the codes and the associated timing offset.

Notation: Lowercase boldface letters denote vectors. The k -th component of a vector \mathbf{x} is denoted by $[\mathbf{x}]_k$. Uppercase boldface letters are used to represent matrices. For a matrix \mathbf{A} , we use $[\mathbf{A}]_{ij}$ to denote the element of \mathbf{A} at its i -th row and j -th column. The \mathbb{Z} denotes the set of all integers. We use $(\mathbf{A})^*$, and $(\mathbf{A})^\top$ to denote complex conjugate transpose and transpose of the matrix respectively. $\mathbf{0}_n$ and $\mathbf{1}_n$ are column vectors of size n with all 0's and 1's respectively.

II. SIGNAL MODEL

A. Single ranging terminal

Consider an uplink OFDMA system with N subcarriers. Thus, each OFDM symbol contains N data symbols. In addition, an OFDM symbol must contain a cyclic prefix. Let the length of the cyclic prefix be N_g . Therefore, the length of a OFDM symbol is

$$\bar{N} = N + N_g.$$

Now consider a particular transmitter T in this system. Let $u(k)$, $k \in \mathbb{Z}$ be the sequence of the channel symbols transmitted by T. These channel symbols are grouped in OFDM symbols. Without any loss of generality we assume that the n th OFDM symbol \mathbf{u}_n consists of $u(k)$, $k = n\bar{N}, n\bar{N}+1, \dots, n\bar{N}+\bar{N}-1$. The n -th OFDM symbol vector is written compactly as

$$\mathbf{u}_n = [u(n\bar{N}) \ u(n\bar{N}+1) \ \dots \ u(n\bar{N}+\bar{N}-1)]^\top. \quad (1)$$

Let us denote the impulse response coefficients between the transmitter T and the base station as $h(p)$, $p = 0, 1, \dots, P-1$. Consequently, the contribution $v(k)$, $k \in \mathbb{Z}$ of transmitter T in the signal received by the base station is given by

$$v(k) = \sum_{p=0}^{N-1} h_p u(k-p-d). \quad (2)$$

In practice, the value of N is more than 200, and typically $h(p) = 0$ for all $p > 50$. The delay d depends on the distance between T and BS. One purpose of the initial ranging process is to estimate d so that the transmitter can align its transmission with the frame boundaries of the base station. In a cellular communication architecture d cannot be arbitrarily large. It

must be bounded. In IEEE802.16 the cell radius is chosen such that $d < N$.

IEEE802.16 identifies some specific uplink subcarriers as 'ranging subchannels'. In the sequel we assume that there are M subcarriers in the group of ranging subchannels, and denote their indices by $\{j_m : m = 1, 2, \dots, M\}$. The value of M in IEEE802.16 is 144. In IEEE802.16 a 'ranging opportunity' spans over two consecutive OFDM symbols when the 'ranging terminals', who wish start communicating via the base station, can send their ranging codes. The ranging codes must be sent via the 'ranging subchannels'.

Consider a ranging opportunity consisting of the OFDM symbols $n-1$ and n , respectively. Suppose T is a ranging terminal who wants to use this ranging opportunity. According to IEEE802.16, T must construct the OFDM symbols \mathbf{u}_{n-1} and \mathbf{u}_n as follows. First it chooses a column \mathbf{c}_ℓ of a pre-specified $M \times G$ code matrix

$$\mathbf{C} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \dots \ \mathbf{c}_G]$$

uniformly at random. Thus the probability that ℓ is a given integer in the set $\{1, 2, \dots, G\}$ is $1/G$. IEEE802.16 specification defines \mathbf{C} precisely. Hence it is known to the BS and all the transmitters including T. After choosing ℓ the terminal T calculates the numbers

$$s_q = \frac{1}{\sqrt{N}} \sum_{m=1}^M [\mathbf{C}]_{m,\ell} \exp\{i2\pi j_m q/N\}, \quad q = 0, 1, \dots, N-1, \quad (3)$$

where $[\mathbf{C}]_{m,\ell}$ denotes the m th component of \mathbf{c}_ℓ , which is also the element at the m th row and ℓ th column of \mathbf{C} . The operation (3) can be seen as the process of modulating the j_m th subcarrier by $\mathbf{C}_{m,\ell}$, and modulating the non-ranging subcarriers with 0. In practice this computation is carried out using the IFFT algorithm, and is compactly given as

$$\mathbf{s} := [s_0 \ s_1 \ \dots \ s_{N-1}]^\top = \mathbf{F}^* \mathbf{\Theta}^\top \mathbf{c}_\ell, \quad (4)$$

where \mathbf{F} is the $N \times N$ FFT matrix such that

$$[\mathbf{F}]_{km} = \exp\{-2\pi i(k-1)(m-1)/N\}/\sqrt{N}, \quad i = \sqrt{-1}.$$

The matrix $\mathbf{\Theta}$ is an $M \times N$ row selector matrix such that the m th row of $\mathbf{\Theta}$ is the j_m th row of the $N \times N$ identity matrix. Recall that $j_m, m = 1, 2, \dots, M$ are the indices associated with the ranging subchannels. The process of modulating the subcarriers via IFFT is equivalent of pre-multiplication by \mathbf{F}^* in (4). The premultiplication of \mathbf{c}_ℓ by $\mathbf{\Theta}^\top$ implies that only the ranging subcarriers are modulated by the appropriate components of the ranging code \mathbf{c}_ℓ .

Using \mathbf{s} the transmitter constructs the OFDM symbols \mathbf{u}_{n-1} and \mathbf{u}_n as

$$\begin{aligned} \mathbf{u}_{n-1} &= [s_{N-N_g} \ \dots \ s_{N-2} \ s_{N-1} \ s_0 \ s_1 \ \dots \ s_{N-1}]^\top, \\ \mathbf{u}_n &= [s_0 \ s_1 \ \dots \ s_{N-1} \ s_0 \ s_1 \ \dots \ s_{N_g-1}]^\top. \end{aligned} \quad (5)$$

To detect the ranging codes the base station works with the first N samples of the n th received OFDM symbol. According to the standard practice in OFDMA, the BS computes an FFT of these samples and then examines the data received in the ranging subchannels.

We first find the contribution

$$\mathbf{v}_n = [v(n\bar{N}) \ v(n\bar{N} + 1) \ \cdots \ v(n\bar{N} + N - 1)]^\top$$

of T in the first N samples of the n th OFDM symbol received by the base station in terms of \mathbf{s} and

$$\mathbf{h} = [h_0 \ h_1 \ \cdots \ h_{N-1}]^\top. \quad (6)$$

Using (1) and (5) note that both \mathbf{u}_{n-1} and \mathbf{u}_n are linear functions of \mathbf{s} . In particular,

$$u(n\bar{N} - k) = \begin{cases} s(k - N) & -\bar{N} < k \leq -N, \\ s(k) & -N < k \leq 0, \\ s(N - k), & 0 < k \leq N, \\ s(2N - k), & N < k \leq \bar{N}. \end{cases} \quad (7)$$

In the following we denote the circular shift operator by $\downarrow(\cdot)$. For instance, the circularly shifted version of \mathbf{s} by k places is given as

$$\mathbf{s}_{\downarrow(k)} := [s_{N-k} \ \cdots \ s_{N-1} \ s_0 \ s_1 \ \cdots \ s_{N-k-1}]^\top. \quad (8)$$

A few steps of algebra using (2) and (7), and using the fact that $d < N$, we get

$$\mathbf{v}_n = \mathbf{H}\mathbf{s}_{\downarrow(d)}, \quad (9)$$

where \mathbf{H} is an $N \times N$ cyclic matrix

$$\mathbf{H} = [\mathbf{h}_{\downarrow(0)} \ \mathbf{h}_{\downarrow(1)} \ \cdots \ \mathbf{h}_{\downarrow(N-1)}]. \quad (10)$$

Expressions (9) and (10) involving circularly shifted versions of \mathbf{s} and \mathbf{h} are typical of OFDM, and are resulted from the way the OFDM symbols \mathbf{u}_n and \mathbf{u}_{n-1} are constructed in (5). This special construction allows us to exploit the identity [14]

$$\begin{aligned} \mathbf{F}\mathbf{x}_{\downarrow(k)} &= \text{diag}(\mathbf{F}(:, k+1))\mathbf{F}\mathbf{x} \\ &= \text{diag}(\mathbf{F}\mathbf{x})\mathbf{F}(:, k+1), \quad k = 0, 1, \dots, N-1, \end{aligned} \quad (11)$$

satisfied by the FFT matrix \mathbf{F} , and any vector \mathbf{x} . Note that we use the standard MATLAB notation $\mathbf{F}(:, k)$ to denote the k th column of \mathbf{F} , and $\text{diag}(\mathbf{x})$ to denote the diagonal matrix such that $[\text{diag}(\mathbf{x})]_{k,k} = [\mathbf{x}]_k$. Recall that for detecting the ranging codes the BS must compute an FFT of \mathbf{v}_n , and then extract the data received in the ranging subchannels. The FFT of \mathbf{v}_n is $\mathbf{F}\mathbf{v}_n$. Hence, the data received from T in the ranging subchannels is given by premultiplying $\mathbf{F}\mathbf{v}_n$ by the row selector matrix Θ . Using (10) and (11) it follows that

$$\mathbf{F}\mathbf{H} = \text{diag}(\mathbf{F}\mathbf{h})\mathbf{F}.$$

Note that $\hat{\mathbf{h}} := \mathbf{F}\mathbf{h}$ is the FFT of \mathbf{h} . Hence by (9) and (11) it follows that

$$\begin{aligned} \mathbf{F}\mathbf{v}_n &= \mathbf{F}\mathbf{H}\mathbf{s}_{\downarrow(d)} = \text{diag}(\hat{\mathbf{h}}) \mathbf{F}\mathbf{s}_{\downarrow(d)} \\ &= \text{diag}(\hat{\mathbf{h}}) \text{diag}(\mathbf{F}(:, d+1)) \mathbf{F}\mathbf{s} \\ &= \text{diag}(\mathbf{F}\mathbf{s}) \text{diag}(\mathbf{F}(:, d+1)) \hat{\mathbf{h}} \\ &= \text{diag}(\mathbf{F}\mathbf{s})\mathbf{F}\mathbf{h}_{\downarrow(d)}. \end{aligned} \quad (12)$$

It is well-known that \mathbf{F} is an unitary matrix, i.e., $\mathbf{F}^*\mathbf{F} = \mathbf{F}\mathbf{F}^* = \mathbf{I}$. Hence (4) implies $\mathbf{F}\mathbf{s} = \Theta^\top \mathbf{c}_\ell$. Hence the data received by the BS at the ranging subchannels due to transmission of T is given by

$$\Theta \mathbf{F}\mathbf{v}_n = \mathbf{E}_\ell \mathbf{h}_{\downarrow(d)}, \quad \mathbf{E}_\ell = \Theta \text{diag}(\Theta^\top \mathbf{c}_\ell) \mathbf{F}. \quad (13)$$

Each of the matrices $\mathbf{E}_\ell, \ell = 1, 2, \dots, G$ is of size $M \times N$, and is known because \mathbf{c}_ℓ is known.

Typically, we know a number P such that $|h_k| = 0$ for $k \geq P$. At this point we emphasize that OFDM can effectively equalize the inter-symbol interference effects only when $P < N_g$. Thus the existence of the upper bound P is a key assumption in OFDM. In addition, the cell radius gives an upper bound D on d . Hence $d + P < D + N_g$. By construction of $\mathbf{h}_{\downarrow(d)}$, we know only first $D + P$ of its rows are non-zero. Hence it is fine to truncate $\mathbf{h}_{\downarrow(d)}$ to a $D + N_g$ dimensional vector, and thus it is enough to work with only first $D + N_g$ columns of \mathbf{E}_ℓ .

B. Multiple ranging terminals

So far we have considered a single ranging terminal, and in (13) we have quantified its contribution to the data vector received by the BS in the ranging subchannels. In this section, we generalize the analysis for multiple ranging terminals, and account of the receiver noise at the base station.

Suppose that the code \mathbf{c}_ℓ is chosen and transmitted by \tilde{N}_ℓ number of ranging terminals. This means that the total number ranging terminals in the system is $K = \tilde{N}_1 + \tilde{N}_2 + \cdots + \tilde{N}_G$. We emphasize that \tilde{N}_ℓ is a random quantity for a given ℓ . Typically $K < 10$, $G = 256$, and the probability that $\tilde{N}_\ell > 1$ is

$$\text{Prob}\{\tilde{N}_\ell > 1\} = 1 - (1 - 1/G)^K - K(1 - 1/G)^{K-1}(1/G),$$

which is a very small number. Hence the probability that two or more ranging terminals will collide by selecting the same code is very small. However for the sake of generality we do not exclude that possibility.

Let $\mathbf{h}_\ell^{(k)}$ and $d_{\ell,k}$, $k = 1, 2, \dots, \tilde{N}_\ell$ denote the channel impulse response vector and the delay of the k th ranging terminal transmitting the code \mathbf{c}_ℓ . Then by the principle of superposition the data vector \mathbf{y} received by the BS at the ranging subchannels is given by, see (13)

$$\mathbf{y} = \sum_{\ell=1}^G \mathbf{E}_\ell \mathbf{h}_\ell + \mathbf{e}, \quad (14)$$

where \mathbf{e} is the additive receiver noise, and

$$\mathbf{h}_\ell = \begin{cases} \sum_{k=1}^{\tilde{N}_\ell} [\mathbf{h}_\ell^{(k)}]_{\downarrow(d_{\ell,k})}, & \tilde{N}_\ell > 0, \\ 0, & \tilde{N}_\ell = 0. \end{cases} \quad (15)$$

is the combined channel vector for all the ranging terminals transmitting the code \mathbf{c}_ℓ . Note that the power received by the BS corresponding to the ranging code \mathbf{c}_ℓ is given by [2]:

$$\Gamma_\ell = \mathbf{h}_\ell^* \mathbf{h}_\ell.$$

III. ESTIMATION OF RANGING INFORMATION

A. Formal problem Statement

Given \mathbf{y} the signal model in (14) the BS needs to

- 1) Find the set $\mathcal{L} = \{\ell : \Gamma_\ell \neq 0\}$;
- 2) For every $\ell \in \mathcal{L}$ find Γ_ℓ and $d_{\ell,1}$ assuming $\tilde{N}_\ell = 1, \forall \ell \in \mathcal{L}$.

In some rare cases $\tilde{N}_\ell > 1$ for some ℓ . In this case the ranging requests of the users who chose \mathbf{c}_ℓ simultaneously would collide. Nevertheless, as we see later, the BS would detect that \mathbf{c}_ℓ was transmitted among others. A IEEE802.16 base station allocates some small bandwidth corresponding to every detected ranging code. The ranging terminals use this ‘grant’ to transmit their buffer status information, and expect to obtain some adequate amount of bandwidth from the BS to commence data transmission. However, if two ranging terminals, say T1 and T2, chose the same code \mathbf{c}_ℓ for ranging request, then the estimate of Γ_ℓ and $d_{\ell,1}$ obtained by the BS during the ranging process would have no physical meaning. In addition, the bandwidth request from T1 and T2 will collide again. Consequently, the BS will not be able to decode the bandwidth request data from T1 and T2. In such a scenario a IEEE802.16 BS does not allocate any further bandwidth corresponding to code \mathbf{c}_ℓ , and after a timeout period T1 and T2 commence the ranging process again [2], [3].

Recall that the first d components of $\mathbf{h}_{\downarrow(d)}$ are zero, see (8). Hence by construction of \mathbf{h}_ℓ in (15), the index of the first nonzero component of \mathbf{h}_ℓ is $1 + \bar{d}_\ell$, where

$$\bar{d}_\ell = \min_{k \in \{1, 2, \dots, \tilde{N}_\ell\}} d_{\ell,k}, \quad \ell \in \mathcal{L}.$$

Clearly, if $\tilde{N}_\ell = 1$, then $\bar{d}_\ell = d_{\ell,1}$. For this reason we propose to estimate \bar{d}_ℓ as the timing offset corresponding to an $\ell \in \mathcal{L}$. When $\tilde{N}_\ell = 1$ this estimate is consistent with our requirements. On the other hand, if $\tilde{N}_\ell > 1$, this estimate will have no practical relevance, for, as discussed above, BS will reject \mathbf{c}_ℓ in the bandwidth request stage.

B. Sparse recovery framework

Recall that for any ℓ only first

$$N_1 := D + N_g \quad (16)$$

components of \mathbf{h}_ℓ are non-zero. Hence

$$\mathbf{E}_\ell \mathbf{h}_\ell = \mathbf{E}_\ell(:, N_1) \mathbf{h}_\ell(1 : N_1).$$

Note that we use Matlab notation $\mathbf{E}_\ell(:, 1 : N_1)$ to denote the submatrix of \mathbf{E}_ℓ formed by taking its first N_1 columns. Similarly, $\mathbf{h}_\ell(1 : N_1)$ denotes the vector formed by taking the first N_1 components of \mathbf{h}_ℓ . Then we can write (14) as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \quad (17)$$

where

$$\mathbf{x} := [\mathbf{h}_1^\top(1 : N_1) \quad \mathbf{h}_2^\top(1 : N_1) \quad \dots \quad \mathbf{h}_G^\top(1 : N_1)]^\top, \\ \mathbf{A} = [\mathbf{E}_1(:, 1 : N_1) \quad \mathbf{E}_2(:, 1 : N_1) \quad \dots \quad \mathbf{E}_G(:, 1 : N_1)].$$

Note that by definition of \mathbf{E}_ℓ in (13), \mathbf{A} is a known matrix. On the other hand \mathbf{x} and \mathbf{e} are unknowns. Typically, the total number of ranging terminals $K = \sum_{\ell=1}^G \tilde{N}_\ell \ll G$, implying $\tilde{N}_\ell = 0$ (and therefore $\mathbf{h}_\ell = 0$) for a vast majority of the values $\ell \in \{1, 2, \dots, G\}$. This makes \mathbf{x} very sparse. This observation motivates a sparse recovery framework for solving the ranging problem.

We propose to estimate a sparse vector \mathbf{x} that is consistent with (17). There are many reliable algorithms for solving

such sparse estimation problems [10], [11]. Denote the sparse estimate by $\check{\mathbf{x}}$. Then the BS can extract the required ranging information as follows. Partition $\check{\mathbf{x}}$ into G number of sub-vectors:

$$\check{\mathbf{x}} = [\check{\mathbf{h}}_1^\top \quad \check{\mathbf{h}}_2^\top \quad \dots \quad \check{\mathbf{h}}_G^\top]^\top,$$

where each $\check{\mathbf{h}}_\ell$ is of length N_1 . Then we declare $\ell \in \mathcal{L}$ only if $\|\check{\mathbf{h}}_\ell\| \neq 0$ and the index of the first nonzero component of $\check{\mathbf{h}}_\ell$ leads to an estimate of \bar{d}_ℓ .

C. Background on sparse recovery methods

If $\mathbf{e} = 0$, then the ideal way to reconstruct a sparse \mathbf{x} from \mathbf{y} requires solving

$$\mathbf{x}_* = \arg \min_{\mathbf{v}} \|\mathbf{v}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{v}, \quad (18)$$

where $\|\mathbf{v}\|_0$, which denotes the ℓ_0 norm of a vector \mathbf{v} , is simply the number of non-zero components in \mathbf{v} . Thus the idea is to find \mathbf{x}_* with the smallest number of non-zero components satisfying $\mathbf{y} = \mathbf{A}\mathbf{x}_*$. The unique representation theorem [15] ensures that under mild technical conditions there is a unique \mathbf{x}_* with $\|\mathbf{x}_*\|_0 < M/2$ satisfying $\mathbf{y} = \mathbf{A}\mathbf{x}_*$.

However, (18) is combinatorial in nature [16]. The most popular alternative approach for relaxing (18) is called Basis Pursuit (BP) [17], [18], where the ℓ_0 norm in (18) is replaced by ℓ_1 norm:

$$\mathbf{x}_* = \arg \min_{\mathbf{v}} \|\mathbf{v}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{v}. \quad (19)$$

Here

$$\|\mathbf{v}\|_1 := \sum_{k=1}^{GN_1} |[\mathbf{v}]_k|.$$

BP can be posed as a linear program [17] over second order cones, and can be solved in polynomial time. In addition, it has been shown in [17] that BP recovers the sparsest solution to $\mathbf{y} = \mathbf{A}\mathbf{x}$ with a very high probability.

The above simple ideas can be adapted quite well even when $\mathbf{e} \neq 0$ [10]. However, the existing algorithms for solving (17) are unable to converge to a satisfactory solution within the time-frame available to solve the ranging problem in practice. In the sequel we propose a new approach to overcome this hurdle.

Our approach blends the nice properties of the ℓ_0 and ℓ_1 methods. The so called ℓ_0 approximation methods [19]–[21] are known to converge very fast if initialized sufficiently close to the final solution. But these methods being non-convex, may often get trapped in some local optimal point when initialized far away from the final solution. The ℓ_1 methods being convex, does not have the local optima problem, but typically take a large number of iterations for convergence. Therefore, we aim to start with an ℓ_1 approach and then handover to an ℓ_0 approach when the solution is ‘sufficiently close’. In particular, we introduce a new ℓ_1 norm minimization method that can obtain a rough estimate of \mathbf{x} in only a few iterations, and then handover to a reliable ℓ_0 -approximation algorithm [20], [21].

D. ℓ_1 optimization algorithm

In this paper we propose to solve a special dual of (19). This dual formulation relies on the theory of minimum norm problems in Banach spaces. Given element $\mathbf{u} \in \mathbb{C}^{GN_1}$ we define the infinite norm as

$$\|\mathbf{u}\|_\infty = \max_{k \in \{1, 2, \dots, GN_1\}} |[\mathbf{u}]_k|.$$

Let us define the bilinear form $\langle \cdot, \cdot \rangle$ as

$$\langle \mathbf{v}, \mathbf{u} \rangle := \frac{1}{2}(\mathbf{u}^* \mathbf{v} + \mathbf{v}^* \mathbf{u}).$$

Note that this bilinear form maps $\mathbb{C}^{GN_1} \times \mathbb{C}^{GN_1}$ onto \mathbb{R} . By Hölder's inequality it follows that

$$\langle \mathbf{v}, \mathbf{u} \rangle \leq \|\mathbf{v}\|_1 \|\mathbf{u}\|_\infty, \quad (20)$$

provided that both $\|\mathbf{v}\|_1$ and $\|\mathbf{u}\|_\infty$ exist. In addition, when we have an equality in (20), then we say \mathbf{v} and \mathbf{u} are aligned. The condition for alignment can be verified to be as follows [22]:

Proposition 1: Let

$$\mathcal{K} = \{k : |[\mathbf{u}]_k| = \|\mathbf{u}\|_\infty\}.$$

Then $\langle \mathbf{v}, \mathbf{u} \rangle = \|\mathbf{v}\|_1 \|\mathbf{u}\|_\infty$ only if

- $[\mathbf{v}]_k = 0$, for all $k \notin \mathcal{K}$;
- For all $k \in \mathcal{K}$ it holds that $[\mathbf{v}]_k = \mu_k \text{conj}([\mathbf{u}]_k) / |[\mathbf{u}]_k|$ for some non-negative number $\mu_k \in \mathbb{R}$.

We are now ready to state the key result allowing us to formulate a convenient dual of (19).

Theorem 1: Let us define the sets

$$\mathbb{V} = \{\mathbf{v} : \mathbf{A}\mathbf{v} = \mathbf{y}\}, \quad \mathbb{U} = \{\mathbf{g} : \|\mathbf{A}^* \mathbf{g}\|_\infty \leq 1\}.$$

Then

$$\min_{\mathbf{v} \in \mathbb{V}} \|\mathbf{v}\|_1 = \max_{\mathbf{g} \in \mathbb{U}} \frac{1}{2}(\mathbf{y}^* \mathbf{g} + \mathbf{g}^* \mathbf{y}). \quad (21)$$

In addition, let \mathbf{x}_* be the solution to the optimization problem in the left hand side of (21), and let \mathbf{g}_* be the solution to the optimization problem in the right hand side of (21). Then

$$\langle \mathbf{x}_*, \mathbf{A}^* \mathbf{g}_* \rangle = \|\mathbf{x}_*\|_1 \|\mathbf{A}^* \mathbf{g}_*\|_\infty. \quad (22)$$

Proof: See [22]. ■

Note that \mathbf{g} is of significantly smaller size compared to \mathbf{x} . Computationally it is a lot more economical to solve the dual in the right hand side of (21), and apply the alignment condition (22) to recover \mathbf{x}_* from \mathbf{g}_* . We write the dual problem as

$$\mathbf{g}_* = \arg \max_{\mathbf{g}} \frac{1}{2}(\mathbf{g}^* \mathbf{y} + \mathbf{y}^* \mathbf{g}) \quad (23)$$

$$\text{subject to } \mathbf{g}^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g} \leq 1, \quad i = 1, \dots, GN_1 \quad (24)$$

where the i th column of \mathbf{A} is denoted by \mathbf{a}_i . We wish to solve it via a primal-dual algorithm [23]. Therefore we write the Lagrangian associated with (23)-(24):

$$L(\mathbf{g}, \mathbf{z}) = \frac{1}{2} \{\mathbf{g}^* \mathbf{y} + \mathbf{y}^* \mathbf{g} - \mathbf{g}^* \mathbf{A} \text{diag}(\mathbf{z}) \mathbf{A}^* \mathbf{g} + \mathbf{1}^T \mathbf{z}\} \quad (25)$$

where $\mathbf{1}$ is a GN_1 dimensional vector of all ones, and $\mathbf{z}/2$ is the real-valued vector of Lagrange multipliers. Each component of \mathbf{z} must be non-negative, and we denote it by $\mathbf{z} \geq 0$. Now it is a standard result in the theory of least squares that

$$\arg \max_{\mathbf{g}} L(\mathbf{g}, \mathbf{z}) = [\mathbf{A} \text{diag}(\mathbf{z}) \mathbf{A}^*]^{-1} \mathbf{y}, \quad (26)$$

$$\max_{\mathbf{g}} L(\mathbf{g}, \mathbf{z}) = \frac{1}{2} \{\mathbf{1}^T \mathbf{z} + \mathbf{y}^* [\mathbf{A} \text{diag}(\mathbf{z}) \mathbf{A}^*]^{-1} \mathbf{y}\}. \quad (27)$$

Therefore, we can obtain \mathbf{z}_* by solving the Lagrangian dual of (23)-(24):

$$\mathbf{z}_* = \arg \max_{\mathbf{z}} \frac{1}{2} \{\mathbf{1}^T \mathbf{z} + \mathbf{y}^* [\mathbf{A} \text{diag}(\mathbf{z}) \mathbf{A}^*]^{-1} \mathbf{y}\} \quad (28)$$

subject to $\mathbf{z} \geq 0$.

In reality, a primal-dual algorithm would solve (23)-(24) and its Lagrangian dual (28) together by finding the solution to the Karush-Kuhn-Tucker (KKT) conditions

$$\mathbf{y} = \mathbf{A} \text{diag}(\mathbf{z}_*) \mathbf{A}^* \mathbf{g}_*, \quad (29)$$

$$[\mathbf{z}_*]_i \geq 0, \quad i = 1, \dots, GN_1, \quad (30)$$

$$(1 - \mathbf{g}_*^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g}_*) \geq 0, \quad i = 1, \dots, GN_1, \quad (31)$$

$$[\mathbf{z}_*]_i (1 - \mathbf{g}_*^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g}_*) = 0, \quad i = 1, \dots, GN_1. \quad (32)$$

Equation (29) follows from (26). The inequalities (30) and (31) must hold because the constraints in (24) and (28) must hold. Equation (32) is the complementary slackness condition which says that for any i either $(1 - \mathbf{g}_*^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g}_*) = 0$ or $[\mathbf{z}_*]_i = 0$. These relations can be used to calculate \mathbf{x}_* from \mathbf{z}_* and \mathbf{g}_* .

Proposition 2: The optimal solution \mathbf{x}_* of (19) is given in terms of \mathbf{z}_* and \mathbf{g}_* as

$$\mathbf{x}_* = \text{diag}(\mathbf{z}_*) \mathbf{A}^* \mathbf{g}_*. \quad (33)$$

Proof: Note that by setting \mathbf{x}_* as in (33) we do satisfy $\mathbf{y} = \mathbf{A} \mathbf{x}_*$. It remains to verify the alignment condition (22).

Now, it must hold that $\|\mathbf{A}^* \mathbf{g}_*\|_\infty = 1$. This is because if $\|\mathbf{A}^* \mathbf{g}_*\|_\infty < 1$, then we could always multiply \mathbf{g}_* by a suitable real valued scalar $\kappa > 1$ such that $\|\mathbf{A}^* (\kappa \mathbf{g}_*)\|_\infty = 1$, and

$$\mathbf{g}_*^* \mathbf{y} + \mathbf{y}^* \mathbf{g}_* < (\kappa \mathbf{g}_*)^* \mathbf{y} + \mathbf{y}^* (\kappa \mathbf{g}_*),$$

leading to a contradiction.

Let us define

$$\mathcal{K} = \{i : \mathbf{g}_*^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g}_* = 1\}.$$

Note that \mathcal{K} is nonempty since $\|\mathbf{A}^* \mathbf{g}_*\|_\infty = 1$. However, the complementary slackness condition (32) implies that $[\mathbf{z}_*]_i = 0$ for all $i \notin \mathcal{K}$. Hence

$$\|\mathbf{x}_*\|_1 = \sum_{i \in \mathcal{K}} [\mathbf{z}_*]_i |\mathbf{a}_i^* \mathbf{g}_*| = \sum_{i \in \mathcal{K}} [\mathbf{z}_*]_i. \quad (34)$$

The last equality follows because by definition of \mathcal{K} we have $|\mathbf{a}_i^* \mathbf{g}_*| = 1$ for all $i \in \mathcal{K}$. Then

$$\mathbf{g}_*^* \mathbf{A} \mathbf{x}_* = \sum_{i \in \mathcal{K}} \mathbf{g}_*^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g}_* [\mathbf{z}_*]_i = \sum_{i \in \mathcal{K}} [\mathbf{z}_*]_i = \|\mathbf{x}_*\|_1$$

Hence we can verify that the alignment condition

$$\begin{aligned}\langle \mathbf{x}_*, \mathbf{A}^* \mathbf{g}_* \rangle &= \frac{1}{2}(\mathbf{g}_*^* \mathbf{A} \mathbf{x}_* + \mathbf{x}_*^* \mathbf{A}^* \mathbf{g}_*) = 1 \times \|\mathbf{x}_*\|_1 \\ &= \|\mathbf{A}^* \mathbf{g}_*\|_\infty \|\mathbf{x}_*\|_1\end{aligned}$$

holds, and thereby the proof is complete. ■

We wish to find a numerical method to solve the KKT equations (29)-(32), which are nonlinear simultaneous equations in \mathbf{z}_* and \mathbf{g}_* . To solve the KKT equations using the primal-dual method, one can relax the complementary slackness condition in (32) to

$$[\mathbf{z}_*]_i (1 - \mathbf{g}_*^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g}_*) = \mu, \quad i = 1, \dots, GN_1. \quad (35)$$

where $\mu > 0$. The value of μ decreases as we progress through the iterations of the primal-dual algorithm. The standard way to handle the modified KKT equations (29)-(31) and (35) is to use the Newton's approach. A derivation of the primal-dual algorithm can be found in [23], and in our case it reduces to the form summarized in Table-I, where we define the function

$$f_i(\mathbf{g}) = \mathbf{g}^* \mathbf{a}_i \mathbf{a}_i^* \mathbf{g} - 1, \quad \text{for } i = 1, 2, \dots, GN_1 \quad (36)$$

with $f(\mathbf{g}) = [f_1(\mathbf{g}) \ f_2(\mathbf{g}) \ \dots \ f_{GN_1}(\mathbf{g})]^T$ and vectors \mathbf{q} and \mathbf{b} such that

$$\begin{aligned}\mathbf{q} &= \mathbf{A}^* \mathbf{g} \\ [\mathbf{b}]_i &= \frac{1}{f_i(\mathbf{g})}, \quad \text{for } i = 1, \dots, GN_1.\end{aligned} \quad (37)$$

The Jacobean matrix of $f(\mathbf{g})$ turns out to be

$$\mathbf{J} = [\mathbf{a}_1 \mathbf{a}_1^* \mathbf{g} \ \mathbf{a}_2 \mathbf{a}_2^* \mathbf{g} \ \dots \ \mathbf{a}_{GN_1} \mathbf{a}_{GN_1}^* \mathbf{g}]^T = \text{diag}(\mathbf{q})^* \mathbf{A}^*. \quad (38)$$

The algorithm in Table-I terminates when a rough estimation of \mathbf{x} has been obtained. We use the following procedure at every iteration to check whether the algorithm has yield a sparse enough estimate of \mathbf{x} . Note that \mathbf{x} is sparse and hence most of its energy should be concentrated on a few number of its components. In fact, it has been shown in [15] that any sparse recovery algorithm can perform well when the energy of \mathbf{x} concentrates within $M/2$ number of its components. At Step-5 in Table-I, we compute $\hat{\mathbf{x}}$. Let $\hat{\mathbf{x}}$ be the thresholded vector constructed from $\hat{\mathbf{x}}$ by retaining $M/2$ most significant components of $\hat{\mathbf{x}}$ while setting others to zero. Now compute

$$\kappa = \frac{\|\hat{\mathbf{x}}\|_2^2}{\|\hat{\mathbf{x}}\|_2^2}.$$

The value of κ can be used as an indicator to measure the proximity of $\hat{\mathbf{x}}$ to the actual \mathbf{x} , while $\kappa \rightarrow 1$ indicates that $\hat{\mathbf{x}}$ may be very close to \mathbf{x} . However, the algorithm in Table-I targets to produce only a rough estimate of \mathbf{x} and hence we do not wait until $\kappa = 1$. In particular, the termination value of κ trades-off computational complexity with estimation accuracy. A larger terminating κ brings $\hat{\mathbf{x}}$ closer to \mathbf{x} for increased number of iterations. The value of κ is being an interesting design parameter. We have observed that the primal-dual algorithm can achieve a $\kappa \geq 0.6$ in only 4 – 5 iterations. The reason behind taking small number of iterations may be due to the formulation of the primal direction search step i.e.,

TABLE I
PRIMAL-DUAL ALGORITHM

Initialization

1. Set $\mathbf{g} = \mathbf{0}$, $\mathbf{z} = \mathbf{0}$, and parameters $\mu, 0 < \alpha \leq 1$.

repeat

2. Compute primal-dual search directions:

$$\Delta \mathbf{g} = (\mathbf{A} \text{diag}(\mathbf{z}) \mathbf{A}^* - \mathbf{J}^* \mathbf{S} \mathbf{J})^{-1} (-\mathbf{y} + \mu \mathbf{J}^* \mathbf{b})$$

$$\Delta \mathbf{z} = -(\mathbf{z} + \mu \mathbf{b} + \mathbf{S} \mathbf{J} \Delta \mathbf{g})$$

$$\text{where, } \mathbf{S} = \text{diag}(\text{diag}(f(\mathbf{g}))^{-1} \mathbf{z}).$$

3. Find $0 < s \leq 1$ such that:

$$3a. f_i(\mathbf{g} + s \Delta \mathbf{g}) \leq 0, \mathbf{z}(i) + s \Delta \mathbf{z}(i) \geq 0; \forall i.$$

3b. The norm of residuals has decreased sufficiently:

$$\|\tau_\mu(\mathbf{g} + s \Delta \mathbf{g}, \mathbf{z} + s \Delta \mathbf{z})\|_2 \leq (1 - \alpha s) \|\tau_\mu(\mathbf{g}, \mathbf{z})\|_2.$$

4. Set $\mathbf{g} = \mathbf{g} + s \Delta \mathbf{g}$, $\mathbf{z} = \mathbf{z} + s \Delta \mathbf{z}$.

5. Compute $\hat{\mathbf{x}} = \text{diag}(\mathbf{z}) \mathbf{A}^* \mathbf{g}$.

6. Set $\mu = \alpha \mu$.

until (A rough estimation of \mathbf{x} has not been obtained)

$\Delta \mathbf{g}$ (see Step 2, Table-I). The popular non-convex iterative re-weighted least square (IRLS) based algorithms like FOCUSS [15], ISL0 [20] use similar formulations to update the estimate of actual sparse signal (\mathbf{x}) in every iterate. It is well known that if IRLS algorithms can avoid local minima then they provide close estimate of actual signal in a small number of iterations.

E. Smoothed ℓ_0 minimization [20]

We use the rough estimate of \mathbf{x} obtained by the primal-dual method to initialize the improved smoothed ℓ_0 (ISL0) algorithm [20]. The ISL0 algorithm is described below. Define the Gaussian functions,

$$f_\sigma(\alpha) = e^{-\frac{\alpha^2}{2\sigma^2}}. \quad (39)$$

Then it is readily verified [20] that, as $\sigma \rightarrow 0$, the function

$$F_\sigma(\mathbf{x}) = \sum_{t=1}^{GN_1} f_\sigma([\mathbf{x}]_t) \quad (40)$$

behaves like $GN_1 - \|\mathbf{x}\|_0$, motivating the following approximate reformulation of (18):

$$\bar{\mathbf{x}}_*(\sigma) := \arg \min_v -F_\sigma(v), \quad \text{subject to } \mathbf{y} = \mathbf{A}v, \quad (41)$$

while taking $\sigma \rightarrow 0$. Like $\|\mathbf{v}\|_0$, the function $F_\sigma(v)$ has many local minima for a small σ . Hence, one solves (41) for a large σ initially, and successively decrease σ using a small factor and solve (41) repetitively. Finally solves (41) for $\sigma = \sigma_0$, where σ_0 is a small positive number. The work in [20] proposed a systematic way to choose σ_0 . In presence of noise, i.e. when $\mathbf{e} \neq 0$ in (17), the following optimization for ISL0 has been considered in [12]:

$$\mathbf{x}_*(\sigma) = \arg \min_v L_\sigma(v), \quad (42)$$

$$L_\sigma(v) := -F_\sigma(v) + \frac{\lambda}{2} \|\mathbf{y} - \mathbf{A}v\|_2^2,$$

where $\lambda > 0$ depends on noise level. A Gauss-Newton type convex-concave procedure is used in [12] to minimize L_σ for

TABLE II
ISL0 ALGORITHM

Input: $\tilde{\mathbf{x}}^{(0)}, \sigma_{st}$

Initialization

1. Set $\sigma = \sigma_{st}$, $\lambda \in [1, 100]$ and $\rho, \eta, \gamma \in [0, 1]$,
 $i = 0, \sigma_0 \in [0.1, 10^{-4}]$.

repeat

2. Set $\beta = 1$.
3. **while** $L_\sigma\{\beta\zeta(\tilde{\mathbf{x}}^{(i)}) + (1 - \beta)\mathbf{x}^{(i)}\} > L_\sigma(\tilde{\mathbf{x}}^{(i)})$
 $\beta = \gamma\beta$.
end
4. $\mathbf{x}^{(i+1)} = \beta\zeta(\tilde{\mathbf{x}}^{(i)}) + (1 - \beta)\tilde{\mathbf{x}}^{(i)}$. Set $i = i + 1$.
5. If $\|\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(i-1)}\|_2 < \eta\sigma$ then $\sigma = \rho\sigma$.

while $\sigma \geq \sigma_0$.

a fixed σ . A detailed description of convergence properties of (41) and (42) can be found in [12], [20]. In particular, the following Lemma gives a direction to minimize L_σ for a fixed σ .

Lemma 1: [12] Define the mapping $\zeta : \mathbb{C}^{GN_1} \rightarrow \mathbb{C}^{GN_1}$ such that

$$\zeta(\tilde{\mathbf{x}}) = \lambda [W_\sigma(\tilde{\mathbf{x}})/\sigma^2 + \lambda \mathbf{A}^* \mathbf{A}]^{-1} \mathbf{A}^* \mathbf{y}, \quad (43)$$

where $W_\sigma(\tilde{\mathbf{x}})$ is a diagonal matrix:

$$W_\sigma(\tilde{\mathbf{x}}) = \begin{bmatrix} f_\sigma([\tilde{\mathbf{x}}]_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_\sigma([\tilde{\mathbf{x}}]_{GN_1}) \end{bmatrix}. \quad (44)$$

Then $\mathbf{x}_*(\sigma) = \zeta\{\mathbf{x}_*(\sigma)\}$. In addition, for any \mathbf{u} there exists a real-valued scalar $\xi \geq 0$ such that

$$L_\sigma\{\kappa\zeta(\mathbf{u}) + (1 - \xi)\mathbf{u}\} \leq L_\sigma(\mathbf{u}). \quad (45)$$

Lemma 1 reveals the fact that $\mathbf{x}_*(\sigma) = \zeta\{\mathbf{x}_*(\sigma)\}$, and motivates a fixed-point iteration approach to find $\mathbf{x}_*(\sigma)$ by solving the equation $\tilde{\mathbf{x}} = \zeta\{\tilde{\mathbf{x}}\}$. Furthermore, the Lemma provides a direction such that $L_\sigma(\tilde{\mathbf{x}})$ is decreasing along $\zeta(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}$.

The ISL0 algorithm is given in Table-II. Here $\tilde{\mathbf{x}}^{(i)}$ denotes the value of $\tilde{\mathbf{x}}$ updated at the i th iteration. The procedure of choosing $\tilde{\mathbf{x}}^{(0)}$ and σ_{st} will be described in the next section. The value of λ in (42) controls the distance between \mathbf{y} and $\mathbf{A}\mathbf{x}_*(\sigma)$. A small value of λ allows $\|\mathbf{y} - \mathbf{A}\mathbf{x}_*(\sigma)\|_2^2$ to be larger. Note that according to (17), the term $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ is equal to the power of measurement noise. Hence, we should choose a small value for λ when noise variance is larger. A procedure for choosing the value of λ is described in [24]. The popular choice of $\lambda = 0.1\|\mathbf{A}^* \mathbf{y}\|_\infty$. For minimizing L_σ , we use $\zeta(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}$ as the descent direction. The γ is a standard backtracking line search parameter [23]. The inner-iteration for minimizing L_σ for a given σ terminates when $\|\tilde{\mathbf{x}}^{(i+1)} - \tilde{\mathbf{x}}^{(i)}\|_2 < \eta\sigma$, (see Step 5). We then update $\sigma = \rho\sigma$. The work in [20] describes a procedure for choosing the values of η and ρ . In particular, it has been shown that the ISL0 remains insensitive to the value of the parameters if we choose $\eta \in [0.1, 0.7]$ and $\rho \in [0.2, 0.9]$. In this work, we set $\eta = 0.5$ and $\rho = 0.3$. The stopping criterion of ISL0 is based on a

small value of σ denoted by σ_0 which depends on the noise level. A wide range of numerical simulations in noisy cases (5 dB to 20 dB SNR) suggest that $\sigma_0 = 0.001$ is good choice.

F. The Handover algorithm

The proposed handover algorithm starts with the ℓ_1 -optimization in Table-I. However, we allow the algorithm to provide only a rough estimation of \mathbf{x} . Once a rough estimate $\hat{\mathbf{x}}$ is obtained it is used as initial $\tilde{\mathbf{x}}^{(0)}$ for ISL0 (see Table-II). The value of σ_{st} for ISL0 can be found in the following way. Assume that $\hat{\mathbf{x}}$ is the minimizer of $L_{\sigma_{st}}(v)$ in (42). Then according to Lemma-1 and (43) we have

$$\begin{aligned} \hat{\mathbf{x}} &= \zeta\{\hat{\mathbf{x}}\} \\ &= \lambda [W_\sigma(\hat{\mathbf{x}})/\sigma_{st}^2 + \lambda \mathbf{A}^* \mathbf{A}]^{-1} \mathbf{A}^* \mathbf{y}. \end{aligned} \quad (46)$$

We need to solve (46) for σ_{st} . However, the equality in (46) may not hold in practice. Then the value of σ_{st}^2 can be approximated by:

$$\sigma_{st}^2 = \arg \min_{\sigma^2} \left\| \frac{W_\sigma(\hat{\mathbf{x}})}{\sigma^2} \hat{\mathbf{x}} - \lambda \mathbf{A}^* (\mathbf{y} - \mathbf{A} \hat{\mathbf{x}}) \right\|_2^2 \quad (47)$$

The optimization problem is non-convex, but one dimensional. An interior trust region algorithm [25] has been applied to estimate σ_{st} from (47) where the initial value of σ is set to $\max_j(|[\hat{\mathbf{x}}]_j|)$ (see [12] for justification).

Suppose the final output obtained from ISL0 is $\bar{\mathbf{x}}$. Due to noise contribution in (17), the estimate $\bar{\mathbf{x}}$ may not exact copy of \mathbf{x} , and hence $\bar{\mathbf{x}}$ will have spurious peaks. As a result, $\bar{\mathbf{x}}$ needs thresholding to perform the code detection. We shall develop a thresholding procedure in the Section-III-H. Let $\check{\mathbf{x}}$ be the thresholded vector constructed from $\bar{\mathbf{x}}$. The BS can detect the active ranging codes and corresponding timing offsets from $\check{\mathbf{x}}$ by using the procedure described in Section-III-B. To obtain the estimate of channel impulse response (CIP), let us partition $\bar{\mathbf{x}} = [\bar{\mathbf{h}}_1^\top \bar{\mathbf{h}}_2^\top \cdots \bar{\mathbf{h}}_G^\top]^\top$. The estimate of CIP corresponding to the ℓ -th active code is $\bar{\mathbf{h}}_\ell$.

G. Computational Complexity Analysis

In this section, we shall analyse computational complexity of the ISL0 algorithm. A similar procedure can be followed to analyze the complexity of the ℓ_1 algorithm in Table-I. As can be seen in Lemma-1, the major fraction of the computation for ISL0 is involved in computing $\zeta(\tilde{\mathbf{x}})$ in (43). However using the matrix inversion lemma in (43) one can verify that

$$\zeta(\tilde{\mathbf{x}}) = W_\sigma^{-1}(\tilde{\mathbf{x}}) \mathbf{A}^* [\mathbf{I}/(\lambda\sigma^2) + \mathbf{A} W_\sigma^{-1}(\tilde{\mathbf{x}}) \mathbf{A}^*]^{-1} \mathbf{y}. \quad (48)$$

We demonstrate a procedure to compute $\zeta(\tilde{\mathbf{x}})$ in (48) efficiently by using FFT. Let us rewrite (48) as

$$\begin{aligned} \zeta(\tilde{\mathbf{x}}) &= W_\sigma^{-1}(\tilde{\mathbf{x}}) \mathbf{A}^* \hat{\mathbf{z}} \\ \text{where, } \mathbf{y} &= [\mathbf{R} + \mathbf{I}/(\lambda\sigma^2)] \hat{\mathbf{z}}, \\ \mathbf{R} &= \mathbf{A} W_\sigma^{-1}(\tilde{\mathbf{x}}) \mathbf{A}^* \end{aligned} \quad (49)$$

Now partition $\tilde{\mathbf{x}}$ into G number of sub-vectors:

$$\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_1^\top \tilde{\mathbf{x}}_2^\top \cdots \tilde{\mathbf{x}}_G^\top]^\top$$

where length of each \tilde{x}_i is N_1 . Construct the matrix $\hat{\mathbf{F}}$ by extracting first N_1 columns of the Fourier matrix \mathbf{F} in (4). We calculate (49) by using the following steps.

- At first we compute $\mathbf{R} = \mathbf{A}W_\sigma^{-1}(\tilde{\mathbf{x}})\mathbf{A}^*$. Since $W_\sigma(\tilde{\mathbf{x}})$ is a diagonal matrix, it follows using (17) and (13) that

$$\mathbf{A}W_\sigma^{-1}(\tilde{\mathbf{x}})\mathbf{A}^* = \sum_{g=1}^G [\Theta \text{diag}(\Theta^\top \mathbf{c}_g)] \hat{\mathbf{F}}W_\sigma^{-1}(\tilde{\mathbf{x}}_g) \hat{\mathbf{F}}^* [\Theta \text{diag}(\Theta^\top \mathbf{c}_g)]^\top$$

and

$$\begin{aligned} [\hat{\mathbf{F}}W_\sigma^{-1}(\tilde{\mathbf{x}}_g)\hat{\mathbf{F}}^*]_{k,\ell} &= \sum_{j=1}^{N_1} \mathbf{w}_j(\tilde{\mathbf{x}}_g) e^{-i2\pi(k-\ell)j/N} \\ &= [\hat{\mathbf{F}}\mathbf{w}(\tilde{\mathbf{x}}_g)]_{k-\ell}, \end{aligned} \quad (50)$$

where $\mathbf{w}(\tilde{\mathbf{x}}_g)$ is a vector constructed from the diagonal components of $W_\sigma^{-1}(\tilde{\mathbf{x}}_g)$. We compute $\hat{\mathbf{F}}\mathbf{w}(\tilde{\mathbf{x}}_g)$ via FFT using $N \log_2(N)$ floating point operations. Recall that the entries of the code matrix \mathbf{C} are $\{+1, -1\}$. In addition Θ is a row selector matrix. Hence, a multiplication by $[\Theta \text{diag}(\Theta^\top \mathbf{c}_g)]$ does not require any floating point operation. In fact, constructing

$$\mathbf{R}_g := [\Theta \text{diag}(\Theta^\top \mathbf{c}_g)] \hat{\mathbf{F}}W_\sigma^{-1}(\tilde{\mathbf{x}}_g) \hat{\mathbf{F}}^* [\Theta \text{diag}(\Theta^\top \mathbf{c}_g)]^\top$$

needs to extract a small block of $\hat{\mathbf{F}}W_\sigma^{-1}(\tilde{\mathbf{x}}_g)\hat{\mathbf{F}}^*$, and changing the signs of some entries. Finally, we compute $\mathbf{R} = \sum_{i=1}^G \mathbf{R}_i$ which requires $(G-1)(M+1)M/2$ flops. Hence this step requires $G\{N \log_2(N) + M(M+1)/2\} - M(M+1)/2$ flops in total.

- Calculate $\hat{\mathbf{z}}$ by solving $[\mathbf{R} + \mathbf{I}/(\lambda\sigma^2)]\hat{\mathbf{z}} = \mathbf{y}$. By using Cholesky factorization, this $M \times M$ positive definite system of equations need $O(\frac{1}{3}M^3)$ flops to compute $\hat{\mathbf{z}}$.
- Compute $\mathbf{A}^*\hat{\mathbf{z}}$ in parts, i.e. we partition $\mathbf{A}^*\hat{\mathbf{z}} = [\tilde{\mathbf{z}}_1 \tilde{\mathbf{z}}_2 \cdots \tilde{\mathbf{z}}_G]$, and form

$$\tilde{\mathbf{z}}_g = \hat{\mathbf{F}}^* [\Theta \text{diag}(\Theta^\top \mathbf{c}_g)]^\top \hat{\mathbf{z}}; \quad g = 1, 2, \dots, G \quad (51)$$

by computing the IFFT of $[\Theta \text{diag}(\Theta^\top \mathbf{c}_g)]^\top \hat{\mathbf{z}}$. Recall that forming $[\Theta \text{diag}(\Theta^\top \mathbf{c}_g)]^\top \hat{\mathbf{z}}$ does not require any additional floating point operation. Hence this step requires $O(GN \log_2(N))$ flops.

- Finally, as $W_\sigma^{-1}(\tilde{\mathbf{x}})$ is diagonal, we need GN_1 multiplications to compute $\zeta(\tilde{\mathbf{x}})$.

Thus in total we need $O(2GN \log_2(N) + GM(M+1)/2 + GN_1 + 1/3M^2(M-1.5))$ flops to compute $\zeta(\tilde{\mathbf{x}})$.

H. Thresholding the recovered signal from ISLO

Let the final output obtained from ISLO is $\bar{\mathbf{x}}$. We denote $\mathbf{v} = \bar{\mathbf{x}} - \mathbf{x}$. The vector \mathbf{v} can be viewed as the recovery error resulted due to noise. To perform a thresholding of $\bar{\mathbf{x}}$, we analyze the statistical property of \mathbf{v} . Using (43), we can write

$$\bar{\mathbf{x}} = \left[\frac{W_\sigma(\tilde{\mathbf{x}})}{\lambda\sigma^2} + \mathbf{A}^*\mathbf{A} \right]^{-1} \mathbf{A}^*\mathbf{y}. \quad (52)$$

Assuming $\|\mathbf{v}\|_2$ is small, the first order Taylor series expansion of $W_\sigma(\bar{\mathbf{x}})$ around \mathbf{x} is

$$W_\sigma(\bar{\mathbf{x}}) = W_\sigma(\mathbf{x}) - W_\sigma(\mathbf{x}) \text{diag}\left(\frac{\mathbf{x}}{\sigma^2}\right) \text{diag}(\mathbf{v}). \quad (53)$$

Now consider (52),

$$\begin{aligned} \mathbf{A}^*\mathbf{y} &= \left[\frac{W_\sigma(\tilde{\mathbf{x}})}{\lambda\sigma^2} + \mathbf{A}^*\mathbf{A} \right] \bar{\mathbf{x}} \\ &= \left[\frac{W_\sigma(\mathbf{x})}{\lambda\sigma^2} \left(\mathbf{I} - \text{diag}\left(\frac{\mathbf{x}\mathbf{v}}{\sigma^2}\right) - \text{diag}\left(\frac{\mathbf{x}^2}{\sigma^2}\right) \right) + \mathbf{A}^*\mathbf{A} \right] \mathbf{v} \\ &\quad + \left[\frac{1}{\lambda\sigma^2} W_\sigma(\mathbf{x}) + \mathbf{A}^*\mathbf{A} \right] \mathbf{x}. \end{aligned} \quad (54)$$

By ignoring second order terms in \mathbf{v} we have

$$\begin{aligned} &\left[\frac{W_\sigma(\mathbf{x})}{\lambda\sigma^2} \left(\mathbf{I} - \text{diag}\left(\frac{\mathbf{x}^2}{\sigma^2}\right) \right) + \mathbf{A}^*\mathbf{A} \right] \mathbf{v} \\ &= \mathbf{A}^*\mathbf{y} - \left[\frac{1}{\lambda\sigma^2} W_\sigma(\mathbf{x}) + \mathbf{A}^*\mathbf{A} \right] \mathbf{x} \end{aligned} \quad (55)$$

For a small value of σ , we can neglect $W_\sigma(\mathbf{x})\frac{\mathbf{x}}{\sigma^2}$, hence

$$\begin{aligned} \mathbf{v} &= \left[\frac{W_\sigma(\mathbf{x})}{\lambda\sigma^2} \left(\mathbf{I} - \text{diag}\left(\frac{\mathbf{x}^2}{\sigma^2}\right) \right) + \mathbf{A}^*\mathbf{A} \right]^{-1} \mathbf{A}^* \mathbf{e} \\ &= \mathbf{D} \mathbf{e} \end{aligned} \quad (56)$$

where we define $\mathbf{D} = \left[\frac{W_\sigma(\mathbf{x})}{\lambda\sigma^2} \left(\mathbf{I} - \text{diag}\left(\frac{\mathbf{x}^2}{\sigma^2}\right) \right) + \mathbf{A}^*\mathbf{A} \right]^{-1} \mathbf{A}^*$. To compute \mathbf{D} , we need the value of \mathbf{x} which is unknown in priori. Nevertheless, we can use an estimate of \mathbf{x} to compute \mathbf{D} . In this work, we use $\tilde{\mathbf{x}}$ as an estimate of \mathbf{x} . Furthermore, computing \mathbf{D} requires inverting a large size matrix. The computation task can be reduced significantly by applying matrix inversion lemma. Let us define $\mathbf{P} = \frac{W_\sigma(\tilde{\mathbf{x}})}{\lambda\sigma^2} \left(\mathbf{I} - \text{diag}\left(\frac{\tilde{\mathbf{x}}^2}{\sigma^2}\right) \right)$. It can be verified that

$$\mathbf{D} = \mathbf{P}^{-1} \mathbf{A}^* [\mathbf{I} + \mathbf{A} \mathbf{P}^{-1} \mathbf{A}^*]^{-1}. \quad (57)$$

Let us partition the matrix \mathbf{D} such that $\mathbf{D}^\top = [\mathbf{D}^{(1)} \mathbf{D}^{(1)} \dots \mathbf{D}^{(G)}]$ where each $\mathbf{D}^{(i)} \in \mathbb{C}^{M \times N_1}$. Also partition $\mathbf{v} = [\mathbf{v}_1^\top \mathbf{v}_2^\top \dots \mathbf{v}_G^\top]^\top$. Assume that \mathbf{e} is complex Gaussian with zero mean and a covariance matrix $\sigma^2 \mathbf{I}$. Hence, the entries of \mathbf{D} are independent of \mathbf{e} . Then the variable $\|\mathbf{v}_i\|_2^2 = \|[\mathbf{D}^{(i)}]^\top \mathbf{e}\|_2^2$ has a generalized chi-square distribution of order M (assuming $M < N_1$) [27]. The procedure for computing the cumulative distribution function (CDF) of a variable having generalized chi-square distribution has been described in [27], [28]. In this work, the CDF of $\|\mathbf{v}_i\|_2^2$ for a threshold τ_i will be denoted by $\chi(\tau_i, \Lambda^{(i)}, \sigma_e^2)$, where $\Lambda^{(i)}$ is the vector containing the singular values of $(\mathbf{D}^{(i)}[\mathbf{D}^{(i)}]^*)$.

Partition $\mathbf{x} = [\mathbf{x}_1^\top \mathbf{x}_2^\top \dots \mathbf{x}_G^\top]^\top$ such that every $\mathbf{x}_i \in \mathbb{C}^{N_1}$. Define a set $S = \{i : \|\mathbf{x}_i\|_2 \neq 0\}$. Then for a given i consider the two hypotheses: $\mathcal{H}_0 : i \notin S; \mathcal{H}_1 : i \in S$. Note that under \mathcal{H}_0 , the distribution of $\|\bar{\mathbf{x}}_i\|_2^2$ is similar to the distribution of $\|\mathbf{v}_i\|_2^2$. To perform hypothesis test on $\|\bar{\mathbf{x}}_i\|_2^2$, we need to select a threshold parameter τ_i . The procedure for selecting the value of τ_i will be described next. The value of $\|\bar{\mathbf{x}}_i\|_2^2$ is checked against τ_i to take a decision between the two hypothesis:

$$\|\bar{\mathbf{x}}_i\|_2^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} \tau_i \quad (58)$$

The threshold τ_i is fixed to achieve a desired false alarm probability ψ according to

$$\begin{aligned} \psi &= P(\|\bar{\mathbf{x}}_i\|_2^2 > \tau_i | \mathcal{H}_0) \\ &= 1 - \chi(\tau_i, \Lambda^{(i)}, \sigma_e^2) \end{aligned} \quad (59)$$

There are total G number of sub-vectors i.e., $\{\bar{x}_i\}_{i=1}^G$. The overall false alarm probability can be defined:

$$P_{fa} = 1 - (1 - \psi)^G \quad (60)$$

To perform thresholding of \bar{x} , we select a desired false alarm probability P_{fa} first. For the P_{fa} , we can calculate the threshold parameter τ_i for every $\|\bar{x}_i\|_2^2$ by using (59).

IV. SIMULATION RESULTS

A typical $N = 1024$ subcarrier OFDMA system, by following the WiMAX standards [1], [2], has been chosen for the simulation. In the system, the carrier frequency is 5.1 GHz, and the associated sampling interval is $T_s = 89.28$ ns. This corresponds to a subcarrier spacing of 10.94 kHz. Length of the cyclic prefix is 64 samples. Total $M = 144$ subcarriers are reserved for the initial ranging purpose, and the total number of available random codes in matrix \mathbf{C} is 32, i.e. $G = 32$. The modulation pulse is a root-raised-cosine function with a roll-off 0.22 and duration $10T_s$. The channel impulse response has a maximum order $P = 30$, and the wireless cell radius is 2.5 km, hence $D = 186$. Similar to [2], [29], we assume that BS has an approximate knowledge about P_{\max} and we set $N_1 = P_{\max} + D$ in (16). The RTs follows a mixed channel model specified by ITU IMT-2000 standards: Ped-A, Ped-B, and Veh-A. The RTs select the channel models with equal probability. The mobile speed varies in the interval $[0, 5]$ m/s for Ped-A, Ped-B channels, and $[5, 20]$ m/s for Veh-A. Since the ranging signal is used to measure the system performance, the signal to noise ratio is defined as $\text{SNR} = 10 \log_{10} \left(\frac{\sigma_h^2}{\sigma_e^2} \right)$, where σ_h^2 and σ_e^2 are the variances of channel impulse response \mathbf{h} and noise term \mathbf{e} respectively. Four different algorithms are considered for performance comparison. The proposed algorithm will be called ‘‘Handover’’. The other three algorithms are the SMUD [2], SRMD scheme discussed in [9], and the MU-GLRT proposed in [29].

We start by finding a good choice of κ to avoid unnecessary iterations in generating the rough estimate (to be used as the initial guess by ISL0) via the ℓ_1 optimization. From a wide range of simulations with different number of IR users and SNR conditions we found that the performance of Handover remain almost same for $\kappa \geq 0.8$. Hence we recommend setting $\kappa = 0.8$, which is used in all the following cases. Figure 1(a) illustrates the code detection performance of the proposed algorithm for different values of false alarm probability P_{fa} (see (60)). The performance is assessed in terms of success of code detection. Recall that the set of active IR code indices is \mathcal{L} . Let $\hat{\mathcal{L}}$ be the set of code indices detected by an algorithm. The probability that $\mathcal{L} = \hat{\mathcal{L}}$, denoted by P_s , is used to quantify the merit of the algorithm. We consider five different values of P_{fa} for code detection. As can be seen in Figure 1(a), the Handover algorithm provides optimum performance for $P_{fa} = 1e-4$. Hence, we recommend setting $P_{fa} = 1e-4$.

Figure 1(b) shows the code detection performance by different algorithms. The performance of MU-GLRT is worse for larger number of active ranging users compared to other three algorithms, whereas the Handover performs best. Note that at moderate SNR i.e., SNR = 10 dB, the Handover

algorithm can recover 6 ranging users with high probability. The performance of SRMD is average compared to other algorithms. For instance, with 4 users and SNR = 10dB, the code detection probability of MU-GLRT, SRMD, SMUD and Handover are 0.36, 0.93, 0.91 and 0.98 respectively. The performance of MU-GLRT degrades rapidly with decreasing the SNR. Hence, we do not illustrate the result of MU-GLRT for lower SNR. With SNR = 3dB and 4 ranging users, the code detection probability of SRMD, SMUD and Handover are 0.88, 0.87 and 0.97 respectively.

Figure 2 illustrates the accuracy of ranging parameter estimations by different algorithms. We do not compare the result with the MU-GLRT at low SNR as its performance is poor in the simulation environment. As can be seen in Figure 2(a), the MSE of power estimate increases with increasing the number of users. Note that with SNR = 10dB, the SMUD, MU-GLRT and Handover exhibit similar performance for small number of users (i.e, for total users 2 in Figure 2(a)). However, their performance difference increases with increasing the number of users. The MSE of power estimate from SMUD for 2 and 5 users (with SNR = 10 dB) are 0.0033 and 0.031 respectively, whereas the MSE from Handover are 0.0034 and 0.017 respectively. The MSE of the timing estimates for the considered ranging algorithms are shown in Figure 2(b). As can be seen, the Handover algorithm outperforms other algorithms with big margin. For example, with 4 ranging users and SNR = 10dB, the MSE for Handover is 4.9 which is 6.38 and 9.54 for SMUD and SRMD respectively. The MSE increases with increasing the number of users. For example, with 5 ranging users and SNR = 10 dB, the MSE of timing offset estimation by Handover, SMUD and SRMD are 6.24, 9.132 and 12.27 respectively. We see that the SMUD performs better than SRMD at high SNR (10 dB), however SRMD outperforms SMUD at low SNR (i.e, 3 dB). We also compare computational complexity of the proposed algorithm with SMUD. The complexity of the SRMD algorithm has not been analysed in [9], hence we cannot incorporate the result in the figure. With SNR = 10 dB and total active users 6, the Handover and SMUD requires 2.58e7 and 1.64e7 flops in average respectively to resolve the IR request.

V. CONCLUSION

In this work, we explored a formulation of the OFDMA initial ranging parameter estimation problem in a sparse signal representation framework. We started with developing a mathematical model that poses the ranging problem into a sparse signal recovery problem. An efficient procedure has been proposed that blends two different types of sparse recovery algorithms. The resulting algorithm exhibits efficient ranging parameter estimation performance.

REFERENCES

- [1] ‘‘IEEE standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems,’’ *IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004)*, pp. 1–2080, 2009.
- [2] M. Ruan, M. Reed, and Z. Shi, ‘‘Successive multiuser detection and interference cancelation for contention based OFDMA ranging channel,’’ *Wireless Communications, IEEE Transactions on*, vol. 9, no. 2, pp. 481–487, 2010.

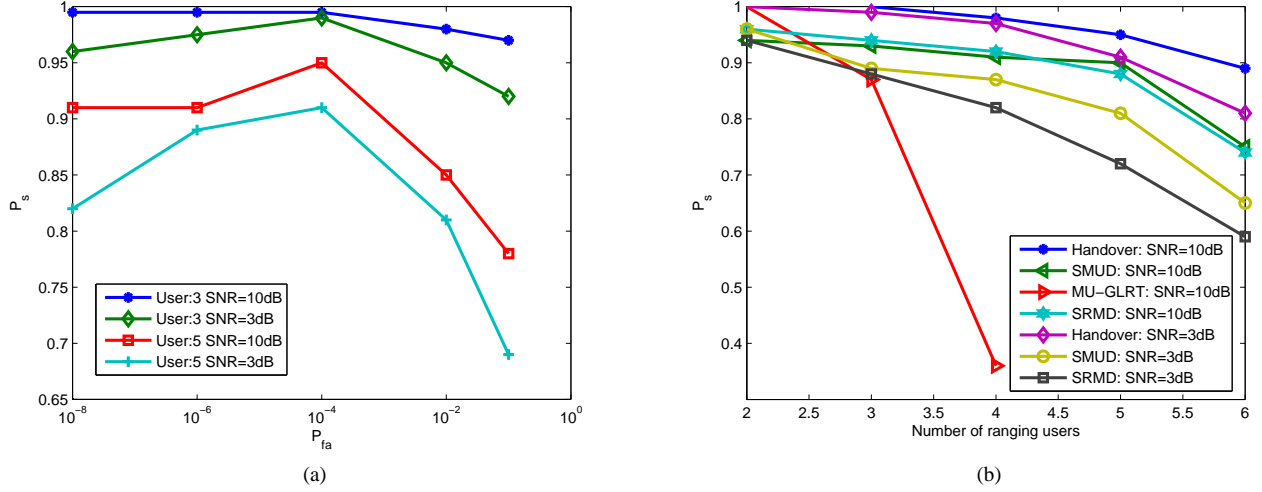


Fig. 1. Active code detection probability by different algorithms. (a) Code detection performance of the Handover algorithm for different values of false alarm probability P_{fa} . (b) Performance comparison of different algorithms, where the value of $P_{fa} = 1e-4$ for Handover algorithm.

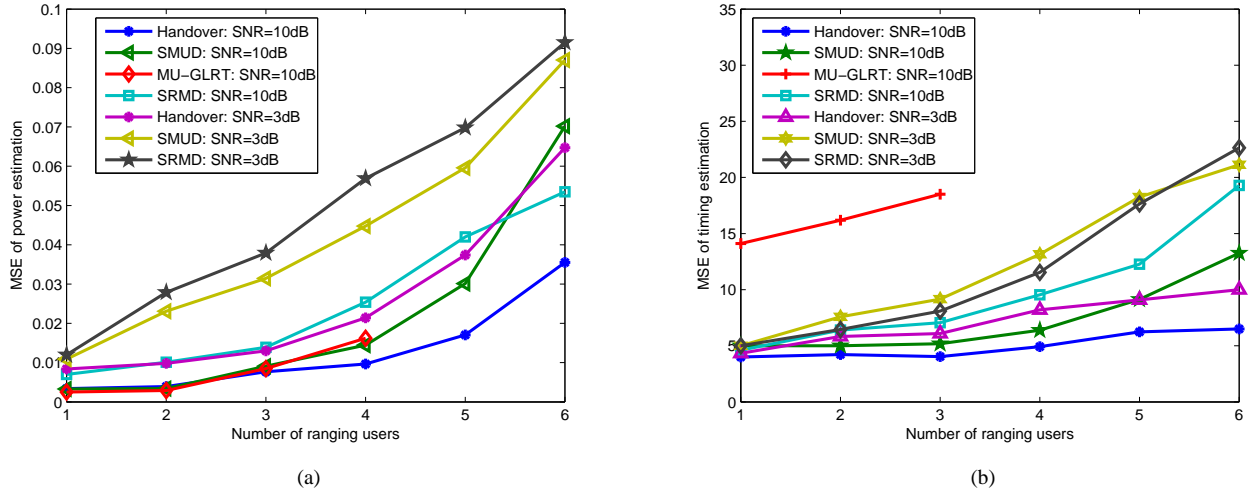


Fig. 2. MSE of ranging parameter estimations. (a) MSE of estimated channel power versus the number of active ranging users, (b) MSE of estimated timing offset versus the number of active ranging users.

- [3] J. K. et al, "Comments on OFDMA ranging scheme described in IEEE 802.16ab-01/01r1," *IEEE 802.16abc-01/24*, 2001.
- [4] D. H. Lee, "OFDMA uplink ranging for IEEE 802.16e using modified generalized chirp-like polyphase sequences," in *Internet, 2005. The First IEEE and IFIP International Conference in Central Asia on*, 2005, pp. 5 pp.-.
- [5] Y. Zhou, Z. Zhang, and X. Zhou, "OFDMA initial ranging for IEEE 802.16e based on time-domain and frequency-domain approaches," in *Communication Technology, 2006. ICCT '06. International Conference on*, 2006, pp. 1-5.
- [6] X. Fu, Y. Li, and H. Minn, "A new ranging method for OFDMA systems," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 2, pp. 659-669, 2007.
- [7] S. Barbarossa, M. Pompili, and G. Giannakis, "Channel-independent synchronization of orthogonal frequency division multiple access systems," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 2, pp. 474-486, 2002.
- [8] V. N. X. Zhuang, K. Baum and M. Cudak, "Ranging enhancement for 802.16e OFDMA PHY," *IEEE C802.16e-04/143*, 2004.
- [9] C.-L. Lin and S.-L. Su, "A robust ranging detection with MAI cancellation for OFDMA systems," in *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, 2011, pp. 937-941.
- [10] E. J. Candès and T. Tao, "Near optimal signal recovery from random projections and universal encoding strategies," *IEEE Transactions on Information Theory*, vol. 52, pp. 5406-5425, Dec. 2006.
- [11] E. Candès and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n ," *Ann. Statist.*, vol. 35, no. 6, pp. 2313-2351, 2007.
- [12] M. Hyder and K. Mahata, "Direction-of-arrival estimation using a mixed $\ell_{2,0}$ norm approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4646-4655, 2010.
- [13] R. Prasad, C. Murthy, and B. Rao, "Joint approximately sparse channel estimation and data detection in OFDM systems using sparse bayesian learning," *Signal Processing, IEEE Transactions on*, vol. 62, no. 14, pp. 3591-3603, July 2014.
- [14] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-time Signal Processing (2nd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.
- [15] I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600-616, Mar 1997.
- [16] D. L. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transac-*

- tions on *Information Theory*, vol. 52, pp. 6–18, Jan. 2006.
- [17] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1999.
 - [18] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, Feb. 2006.
 - [19] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, “A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm,” *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 289–301, Jan. 2009.
 - [20] M. M. Hyder and K. Mahata, “An improved smoothed ℓ^0 approximation algorithm for sparse representation,” *IEEE transactions on Signal Processing*, vol. 58, no. 4, pp. 2194–2205, april 2010.
 - [21] M. Hyder and K. Mahata, “Coherent spectral analysis of asynchronously sampled signals,” *Signal Processing Letters, IEEE*, vol. 18, no. 2, pp. 126–129, 2011.
 - [22] D. G. Luenberger, *Optimization by Vector Space Methods*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1997.
 - [23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
 - [24] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An interior-point method for large-scale ℓ_1 -regularized least squares,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.
 - [25] T. Coleman and Y. Li, “An interior trust region approach for nonlinear minimization subject to bounds,” *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 418–445, 1996.
 - [26] A. Hasan, E. Kerrigan, and G. Constantinides, “Solving a positive definite system of linear equations via the matrix exponential,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 2299–2304.
 - [27] R. B. Davies, “Algorithm as 155: The distribution of a linear combination of χ^2 random variables,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 29, no. 3, pp. 323–333, 1980.
 - [28] J. Sheil and I. O’Muircheartaigh, “Algorithm as 106: The distribution of non-negative quadratic forms in normal variables,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 26, no. 1, pp. 92–98, 1977.
 - [29] L. Sanguinetti and M. Morelli, “An initial ranging scheme for the IEEE 802.16 OFDMA uplink,” *Wireless Communications, IEEE Transactions on*, vol. 11, no. 9, pp. 3204–3215, 2012.